Whitespace

Vehicle Design Report

IEEE Robot Team

University of Wisconsin – Madison

Ben Gartner, Brian Prodoehl, Chris Riley, Mark Schneider, Mike Slutskiy

13th Annual Intelligent Ground Vehicle Competition
Oakland University – Rochester, MI
June 11-13, 2005

Required Faculty Advisor Statement

I certify that the engineering design of the new vehicle, Whitespace, described in this report, has been significant and equivalent to what might be awarded credit in a senior design course.

Dan Botez
Prof. of Elec. and Comp. Engr

1. Introduction

The IEEE Robot Team from the University of Wisconsin – Madison designed and created a new robot to compete in all 3 challenges of the 13th annual Intelligent Ground Vehicle Competition (IGVC). The autonomous robot, named Whitespace, was the tenth robot produced by a dynamic team of students over the last eleven years. With increased agility, mobility, and intelligence, Whitespace represents the team in their second IGVC appearance. With a new robot and strategy, the team has pooled their experience and genuine love of robotics to innovatively improve through last year's lessons.

2. Innovations

Whitespace features many new innovations from bolt to microchip. Starting ground up, the drive system offers a compact, differential steer, two wheel drive system utilizing two 18 volt cordless drills as motors. The frame is built to be weather resistant, compact, and rugged while providing space and protection to all the critical components. Among the critical components, the electronics are simple and elegant with room for future performance upgrades. Outfitted with a camera, GPS unit, and digital compass, Whitespace carries the tools needed for efficient autonomous navigation. Utilizing these tools, Whitespace has powerful software to quickly and efficiently make navigation decisions. Also featured is a "flick-of-switch" option allowing fast switching between autonomous and manual modes.

3. Design Process

To create an innovative robot, our team adhered to the design process in Figure 1.

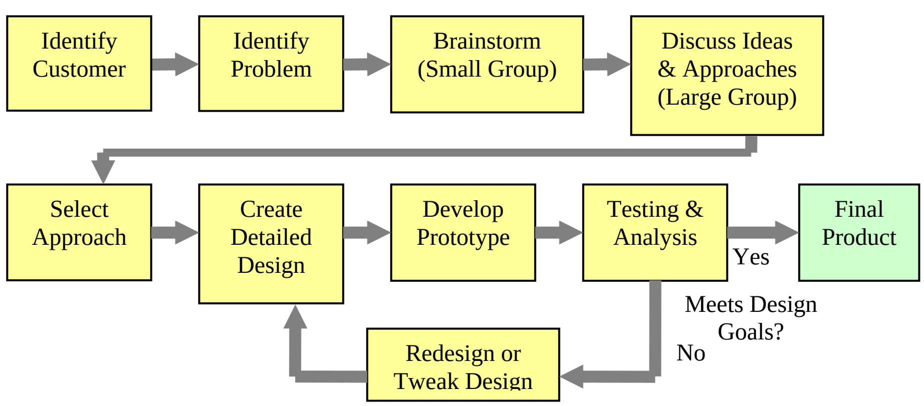


Figure 1: Design Process

3.1. Customer Identification

Beginning our design process, we sought to define our customer base. We concluded that our customers consisted of our team, our advisors and college, our sponsors, the IGVC judges, the IGVC hosts, and the US Military. With each group possessing different motivations, their end goals sum up to three primary goals. These goals are having the students earn valuable skills and experiences; create a high quality product using engineering techniques; and to further the field intelligent, autonomous vehicles.

3.2. Recognition of the Problem

Keeping the customer goals in mind, the team attempted to gain a good grasp of the competition. To accomplish this, we studied this year's rules and previous years design reports. We also discussed what went well or not so well for our team in last year's competition. After examining the problem thoroughly, we created categories in which we wanted to focus our effort.

3.3. Team Organization

Breaking the problem down, the IEEE Robot Team formed three primary groups, a mechanical, an electrical, and a software group. These groups routinely worked together to ensure unity in design, while still maintaining focus on their sections. For each section a few projects were established. These projects developed their own set of goals, where ideas were brainstormed and discussed to choose the best approach. The team members were free to choose the projects that they would like to work on, with regular assistance from the more experienced team members readily available. The team members are listed in the table below with the projects they contributed to.

Member Names	Major	Class	Projects
Ben Gartner	CmpE	Senior	Team Leader and Electrical - Embedded
Brian Prodoehl	EE		Software and Electrical - AI, Vision, Network, Main App., Control, E-Stop
Chris Riley	ME	Senior	Mechanical and Electrical - Chassis, Computer Model, Power System
Mark Schneider	CmpE	Junior	Software and Electrical – AI, Microcontroller, Computer Interface
Mike Slutskiy	Physics	Junior	Software and Mechanical – AI, Chassis

Table 1: IEEE Robot Team Members, 2005 IGVC

3.4. Preliminary Ideas

Through controlled brainstorming sessions, the IEEE Robot Team produced a plethora of approaches from basic robot design to competition strategies. In order to sift through the approaches, our team produced concept charts used to evaluate each idea's advantages and disadvantages. Below, in figure 2, is the concept chart we used while deciding the drive system.

Approach \ Criteria	Stability	Smooth handling	Turning angle	Turning resistance	Turning ability	Tire ware	Propensity to get stuck	Drive train breakage	Simplicity	Ability to model (AI)	Ease of control	Ease of implementation	Resources available	Cost	**Totals**
Skid-steer (datum)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2 Wheel-differential	0	+	0	+	-	+	-	+	0	+	0	0	0	0	+3
Ackerman w/ differential	0	+	-	+	-	+	-	+	-	+	+	-	-	-	-1
Ackerman w/out differential	0	-	-	0	-	+	-	+	-	0	-	-	-	-	-7
Bicycle	-	+	-	+	-	+	-	+	-	+	-	-	-	-	-4
Tricycle	-	+	-	+	-	+	-	+	-	+	+	-	0	0	0
Front and back independent sections	-	+	-	+	-	+	-	+	-	-	-	-	-	-	-6
Rotating 3-wheel base	0	+	0	-	+	+	-	+	-	+	+	-	-	-	0

Figure 2: Concept Chart - Steering/Drive System

In addition to the customer goals we established, we had additional criteria driving our designs. These criteria included cost, ability to obtain resources, and the reliability of the designed systems. These in mind, some of our preliminary design concepts of using a laser range finder and a custom ultrasonic peripheral scanner were passed over. Software choices also were influenced by these criteria. Despite considering developing code with C/C++, the team's experience led to the choice to develop the main algorithms using MATLAB and eventually implementing them in Java.

3.5. Software Development

3.5.1. Methods

Proper software design and coding techniques were followed, resulting in highly modular, fully-encapsulated classes and packages. Unit testing was used where applicable, and all software was written with easy readability and maintenance in mind.

3.5.2. Design Tools

Many design tools were used in the development of Whitespace. Software development was done with the Eclipse IDE for Java and Microchip MPLAB IDE for the microcontrollers. Physical models were designed using SolidWorks 2004. CadSoft Eagle was used to design electrical schematics and produce printed circuit board layouts.

3.6. Construction

Prior to actual construction of the robot, each component was designed and modeled in detail in an effort to minimize manufacturing mistakes, flaws, and part failures. The layout of components and structure of the chassis were modeled using SolidWorks, allowing for critical decisions regarding basic structure and available space to be made before actual manufacturing.

A majority of the necessary parts were machined by students from purchased stock material. With the frame welded together, other parts such as the motors, sprockets, and bearings were mounted. Finally, additional components such as the sensors, camera, and electronics were added to allow the robot to function autonomously.

3.7. Testing

Upon completion, each component was thoroughly tested to ensure safe, reliable, and durable use. These tests became a vital portion in troubleshooting and tweaking of systems and components. See section 9.2 for test result details.

4. Competition Strategy

4.1. Autonomous Challenge

Whitespace, under normal path navigation, operates in a dynamic mode, whereby it deciphers the optimal route using the previous captured image. The mechanism for optimal path planning is explained later in the Vision and AI sections of Software Design. When

Whitespace encounters what it interprets as a dead end trap, it changes mode to locate a solution to overcome the obstacle. By broadening its search pattern it looks for a more appropriate path. If one of these paths are not discovered Whitespace will back track slightly and again search. While this goes on, Whitespace uses previously recorded compass and GPS readings to remain on the path and to ensure that it does not completely turn around and head down the path the wrong way.

4.2. Navigation Challenge

Whitespace implements a waypoint seeking routine for the navigation challenge. Using a current location and a final target waypoint, Whitespace sets up two intermediate waypoints. When an obstacle presents itself in the current drive path, Whitespace alters its intermediate waypoints, plotting a new path around the obstacle. The closer an obstacle in the path becomes, the more drastic the evasion path becomes. Our approach is similar to the wall following concept for maze navigation.

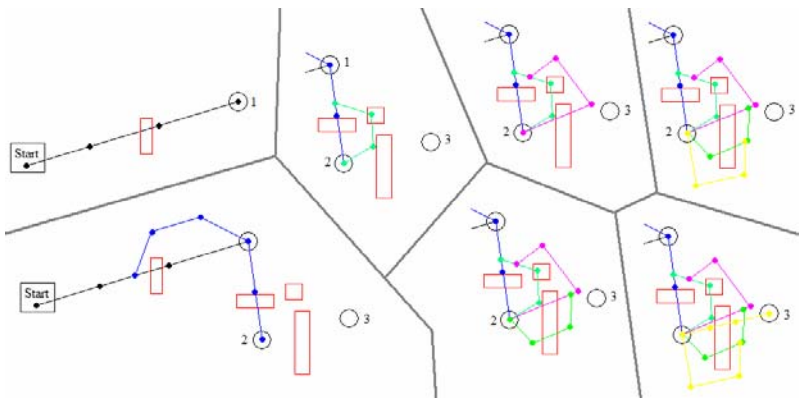


Figure 3: Navigation Challenge Navigation Strategy

5. Mechanical Design

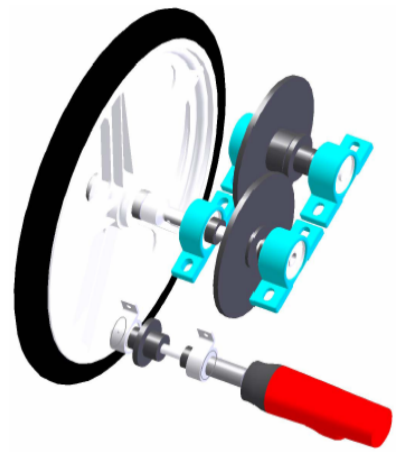
5.1. Chassis

Whitespace's chassis was constructed by using donated angle iron. With the design produced in SolidWorks, special care was given to ensure the necessary components were provided adequate space and accessibility, while maintaining compact.

5.2. Drive System

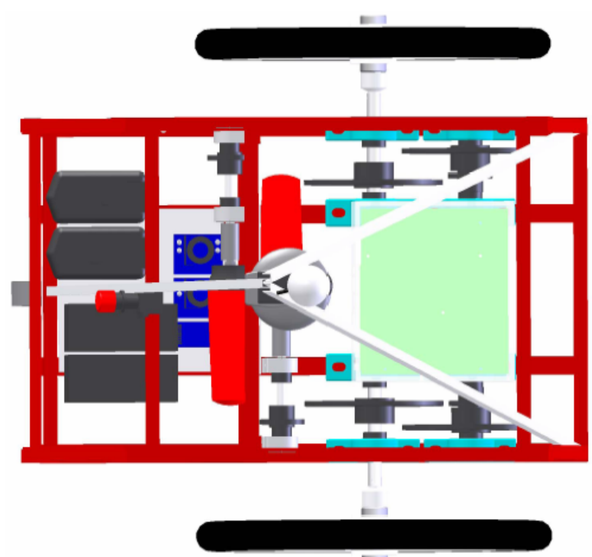
Whitespace's drive system was chosen and designed for a few key reasons. These reasons include adequate stability, minimal turning radius, overall simplicity, and a minimal cost.

The drive system is set up where one motor powers a drive wheel on one side of the vehicle. The motors chosen are two 18 Volt cordless motors from a Milwaukee drill, providing approximately 400 inch-pounds of torque each. The speed of the drill is reduced by a 9:1 ratio using chain and sprockets. The sprocket is then attached to an axel which is locked into a wheel. Each drive wheel is able to output approximately 360 lbf. This force is ample for turning and to provide locomotion for the robot up any inclines presented in the competition. A caster is used in the rear of the robot to stabilize the two drive wheels.



5.3. Sensor Placement

The sensors were place in strategic locations that aided their collection of the type of information they were intended for. The camera was mounted approximately 4.5 feet off the ground facing forward and pointing downward at approximately 40 degrees. This placement can detect objects up to 15 feet away. The GPS unit and compass are located in the center of the robot on the same level of the E-Stop. In this location they are free from interfering signals while being located near the computer.



5.4. Other Component Placement

Several other important components such as the batteries, computer, E-Stop, and payload needed to have a safe accessible location. The two 18V batteries, needed to drive the robot, and the two 12V batteries, powering the computer and sensors, are located in the back and are accessible through doors in the rear panel allowing easy battery swapping. The computer, in the center of the front, can be easily removed once the top panel is opened. The E-Stop button was located 2 feet off the ground, centered on the back of the robot. The payload was placed centered on top of the robot to provide a clear view for its camera.

6. Electrical and Electronics Design

6.1. Overview

Electronics plays a vital role in everything from safety to the durability and reliability of the robot and even on how long the robot can operate. Special care was given to these systems ensuring that all circuitry was carefully laid out and that all wiring met with standards.

6.2. Power System

The primary responsibility of the power system is to safely and efficiently deliver sufficient power to all of the components that need it. To do this we examined the power requirements and added a few safety measures discussed further in the electrical safety section (6.6).

6.2.1. Power Requirements

Below, in table 2, are the power requirements for each system or component.

Item Requiring Power	Req. Voltage	Req. Current	Req. Power	
Computer	12 V	4.3 A	51.6 W	
Logitech QuickCam Pro 4000	Included	with	computer	
GPS-RadioShack DigiTraveler	5 V	75 mA	0.4 W	
Compass-PNI Vector 2x	5 V	10 mA	0.05 W	
LEDs	5 V	0.1 A	0.5 W	
Cordless Mil. Drill Motors	18 V	2.4 A	6.3 W	
Victor 883 Speed Controllers	18 V	10 mA	0.2 W	
Item Supplying Power	Sup. Voltage	Sup. Current	Sup. Power	Est. Life
12V Batteries	12 V	4.4 A	14.4 A-hr	17 min.
18V Batteries	18 V	2.4 A	4.8 A-hr	20 min.

Table 2: Power Requirements

6.2.2. Power Management

In order to monitor the level of the batteries, analog voltage gages, on the back of the robot, have been connected to both the 12 and 18 volt battery sets.

6.2.3. Batteries

The power supplied to the robot comes from two sets of batteries. The first set is two 18V, 2.4A-hr cordless drill batteries from Milwaukee Power Tools. These are used to run the motors and speed controllers. These batteries were chosen because they were the batteries provided with the drills and it only seemed natural to utilize them. The second set is two 12V, 7.2A-hr sealed lead-acid batteries. These batteries power the computer, digital compass, and GPS unit. Each set of batteries have the two batteries connected in parallel in order to provide long battery life. In addition, capacitors were placed between the batteries for safety to prevent sparking while connecting new batteries. The batteries were positioned for easy access to allow quick battery swapping.

6.3. Sensors

6.3.1. Camera

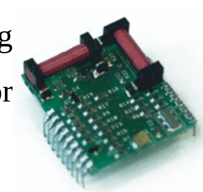
The camera selected by the team is a Logitech QuickCam Pro 4000. This Webcam was chosen because of its compatibility with Linux and USB 2.0. The camera is capable of handling 640x480 video at 30 frames per second.

6.3.2. GPS

The GPS unit the team is using this year is the RadioShack DigiTraveler. The DigitTraveler is able to provide a GPS location with in 1 m, updating once every second.

6.3.3. Compass

The digital compass the team selected is the Vector 2x from PNI. Being inexpensive, low power and having a resolution of 1 degree at 2.5Hz, the Vector 2x is an ideal solution for our needs.



6.4. Computer

The computer the IEEE Robot Team used is custom built. We used a Shuttle MK40V v1.3A mother board running off of a Duron 1.0GHz processor. Instead of a typical hard-drive, the team used a SanDrive 512MB CompactFlash card. To power the computer we use a ETON ET866 DC to DC converter. The computer operates under Linux in order to keep background tasks to a minimum.

6.5. Vehicle Control

Whitespace is controlled through a series of measures. A multi-tiered control system uses the parallel port on the computer to send bits to a microprocessor. The microprocessor creates two PWMs which are outputted with one going to each speed controller. The speed controller takes the signal and provides the motor with another more powerful PWM which the motor uses to turn the wheels. Further explanation is given on how it decides to move in the AI and Control software sections.

6.6. Electrical Safety

Electrical safety was one of our primary concerns. The team took several measures to ensure that Whitespace would be safe both during construction and while in use. The first safety measure is the E-stop. Holding to the rules, we added a push emergency stop button which disengages power to the motors. The motors come to a quick stop due to back-emf produced by the motors themselves. In addition to the push E-stop we have a remote controlled e-stop capable of stopping the robot from a distance of 50 yards. The wiring of Whitespace had some safety measures including: circuit breakers protecting the speed controllers; fuses protecting against voltage spikes; capacitors between batteries protecting against sparks while connecting new batteries; and the wiring standards were closely watched to ensure good connections and wire protection. Weather protection was also a major concern. To protect the electrical and electronic systems from becoming wet, vulnerable components were place inside, where they were sealed from water trickling in or being splashed up by the wheels.

7. Software Design

7.1. Overview

The software was targeted to run on a standard AMD Duron processor under Linux, but care was taken to make sure the software could be developed and tested under Windows. Linux was chosen as an operating system due to its flexible customization that allow for optimization of the OS for low power and small hard disk space. Java was selected as a programming language to provide full platform independence.

7.2. Main Control Application

The main control application links each of the systems together and implements vision, mapping and pathfinding functionality. The control application was designed to allow maximum configurability, allowing the robot to be fine-tuned based on real world performance under varying operational conditions. Because of this, it is possible to test the software in various stages of completeness. A simple test would involve submitting an image to the artificial intelligence and verifying the correctness of the chosen path. A more complete test would involve the robot acquiring images through the onboard camera, converting them into a digital map and attempting to complete the course without breaking competition rules. This approach ensured that often unpredictable artificial intelligence techniques could be verified as legitimate before the rest of the software and hardware development was completed. The control application was implemented in Java and was designed to be platform independent, allowing for maximum flexibility in software and hardware.

7.3. Vision System

The vision system needs to acquire images of the area in front of the robot and transform that image into a segment of a top-down map of the course, in which obstacles are white, and everything else is black. This has always been the main consumer of processor bandwidth and resources, so special care was taken in its design and implementation. To optimize speed and efficiency, a native, compiled solution was sought. The Java Media Framework (JMF) provided a hardware interface that remains consistent regardless of operating system. Because the target machine runs Linux and several of the development machines run Windows, this allowed us to access the camera without using a proprietary interface. The Java Advanced Imaging API (JAI) provided the system libraries capable of performing the complex image filtering and manipulation. JMF provides support for video capture and frame grabbing, while JAI provides all of the image processing operations we need in a fast, compiled format (many of which take advantage of MMX). This solution can easily acquire images, turn them into black-and-white images (with white being obstacles and path edges), and transform them, fulfilling many of the requirements. The only missing link is determining the threshold with which to clean up the image, or the way to determine whether a pixel is part of an obstacle or not. To help find the best way of thresholding the images, many images from previous years (obtained from the competition web site) were analyzed. In particular, histograms were observed. It was concluded that the red and green intensity were not very uniform between the majority of images, but the blue intensity between all images was nearly the same, with the majority of the intensities at very low levels, and a smaller-yet-still significant group at high intensities. This group at high intensities was found to correspond to the path edges, artificial pot-holes and barrels. Handpicking a blue intensity level between the low intensity peak and high intensity peak, and thresholding the image based on that proved successful for all images, but the intensity level needed to be tweaked for each image in order to achieve the best performance. Going back to the histogram, it was determined that the optimum threshold was always right after the main hump declined. An algorithm was quickly developed to smooth the histogram and find the point beyond the maximum at which the slope is within a certain tolerance of a preset value. Applying this algorithm to find the threshold for each image yielded spectacular results. This algorithm is now an integral part of the vision system, and is capable of quickly and effectively determining a new threshold level. We have tested this vision system in many lighting conditions and have had great success.

7.4. Artificial Intelligence

The primary function of the artificial intelligence is to maintain a map of the course, in the form of a probability map, representing which areas are safe to drive on and which are not. The map segments from the vision system and the current positional data of the robot are used to merge the latest frames with the past assumptions. As the map is updated, it is also probed to determine the longest possible forward path. Because of the differential drive system, the robot is highly mobile and well suited for driving along a piecewise linear path. Thus at any point, the artificial intelligence must determine which direction provides the best possible path. This is accomplished by probing the map and finding the longest acceptable straight line path for the robot. The assumption that the longest path represents the correct direction was one of the main focuses of our testing. It was determined that this assumption is largely valid.

Interested mainly in moving forward through the course, it is essential that the artificial intelligence also maintain a layer of directional data pertaining to the course. This directional field is populated with the optimal directions as determined above. When the AI is determining the optimal path, it only looks within a specified range of the directional field. This can lead to situations in which the robot becomes trapped, and these situations are handled by special cases in software. In this case, the robot attempts to back up to the last major decision point, and then makes an alternate path decision.

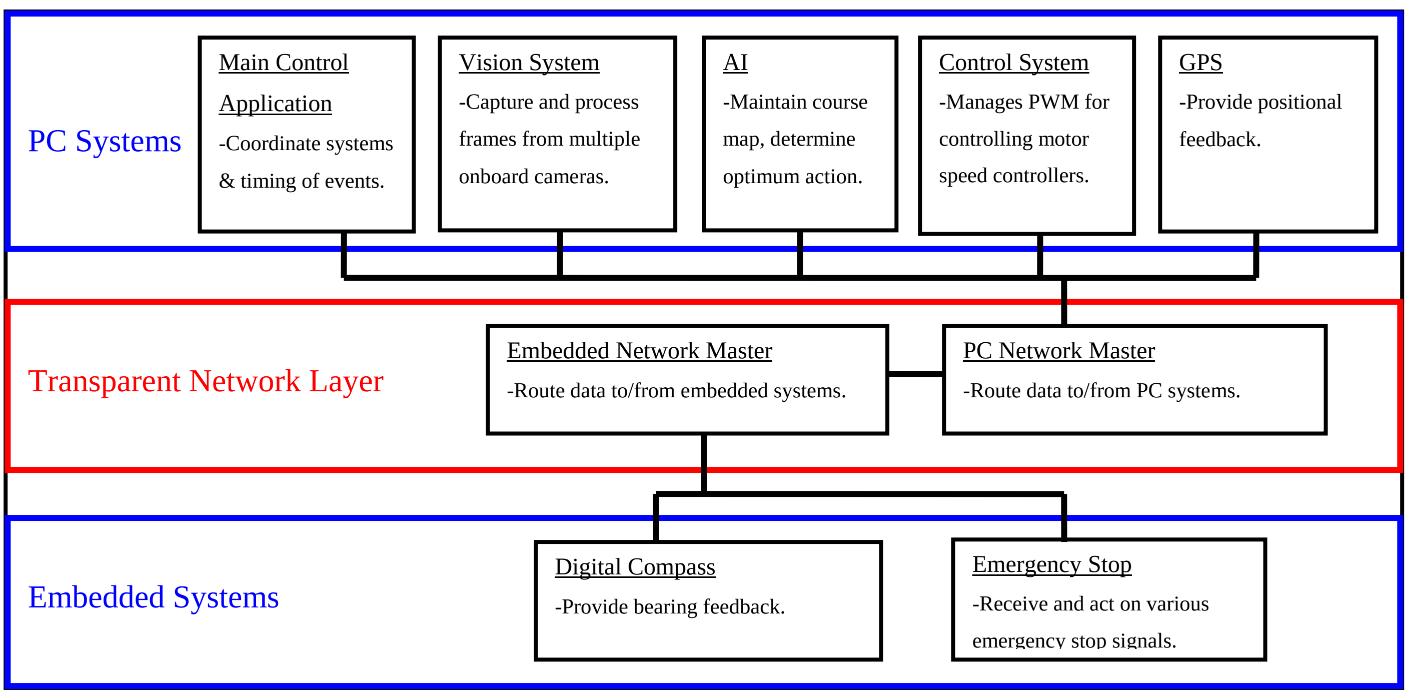
7.5. Control System

The main purpose of the control system is to translate the directional commands of the pathfinding artificial intelligence into the robot’s physical movement. This is accomplished by independently selecting the speed of the two drive wheels. Identical speeds results in straight line forward or reverse motion, while opposite speeds results in rotational movement. Each wheel has a speed resolution of 4 bits, allowing for 7 distinct speed settings per wheel in each direction. These distinct speed settings allow the drive system to ramp up gradually, eliminating jerky motion that leads to unexpected movement.

The software sends movement commands to a separate microcontroller through the parallel port. This microcontroller converts an 8 bit value into two separate pulse width signals, which are respectively sent to the wheels' speed controllers. The speed controllers convert this small voltage signal into a suitable high voltage signal for the drive motors. This system allows for a full range of intended motion.

The digital compass acts as a feedback system to ensure the reliability of intended rotational motion. The intended motion of the robot is based on measured data of turning speed. The compass acts as a sensor to confirm that the robot has performed the intended rotation to a certain degree of accuracy. Accurately positioning the robot not only provides better paths, but allows image data to be more accurately combined into an overhead map.

8. Systems Integration



8.1. Overview

All systems were designed with modularity in mind, so integration was a relatively simple task. The network effortlessly linked the embedded and PC-based systems together, and the PC main control application oversaw the operation of the PC systems. The PC vision, AI, and control systems were designed and developed as fully encapsulated packages. This made testing very simple, and ultimately integrating them into one application was a trivial task.

8.2. Sensor Integration

The sensors were designed so that each served a purpose, and functioned in a way that none of the other sensors could. The camera is able to look far ahead and determine the location of the boundaries and various obstacles. Both the compass and GPS unit provide feedback verifying the predicted performance. The digital compass reports the robot's current bearing, and the GPS is able to report the robot's immediate location.

9. Predicted Performance

9.1. Preliminary Tests

Tests were performed on all systems throughout the development process to ensure safe, reliable execution in any situation. Mechanical and electrical systems were tested verifying predicted speeds, battery life, weather proofing and emergency stops. Software testing entailed both low and high level testing. The low level testing included the polling frequency of the camera, compass, and GPS unit and object detection from captured images. High level testing dealt with obstacle detection range, image processing latency, pathfinding, waypoint navigation, and dead end behavior. These tests were performed under varying temperature, weather and lighting conditions.

9.2. Test Results

Tests	Predicted	Experimental
Max Speed	2.5 mph	2.2 mph
Ramp climbing ability	30 grade	25 grade
Reaction times – (compass, GPS)	(2.5, 1) readings/sec	(1.33, .67) readings/sec
Reaction times – Vision	10 fps, no lag	12.5 fps, 0.5s lag
Reaction times – E-Stop	2sec, 4ft	1sec, 1.75ft
Battery life (18V, 12V)	(20min, 17min)	(23min, 16min)
Obstacle detection distance	15ft	16.5ft
Waypoint accuracy	1m	2.5m

Table 3: Numerical Test Results

Further testing is required for cases involving dead ends, traps and potholes. The vision is also under modification to minimize lag and increase speed.

10. Other Design Considerations

Beyond safety and solid mechanical, electrical and software designs, the team kept in mind a few other key design considerations.

10.1. Reliability and Durability

Reliability and durability was built into all parts of the robot. Reliability within the robot's software and electronics was achieved through rigorous testing of each system and component. Durability of the robot stems from the careful planning of robust systems and components. The drive system was examined in detail to locate any potential flaws and design solutions with safety factors in order to prevent failure. The reliable and durable systems built into Whitespace contribute to its safe, consistent performance.

10.2. Maintenance and Interchangeability

The ability to service and change parts is an important issue. Our team set out about trying to make Whitespace so that it was easy to access and replace all of the key components. Special care was given to components such as batteries and the computer. Batteries were set up to be easily swapped. The computer was design to be easily removed and upgradeable.

10.3. Cost

The cost of creating a robot from scratch can be a substantial investment. Many of the parts and material our team used for Whitespace were either donated or on loan.

Category	Item	Qty.	Price 1st	Price 2nd	Retail
Computer	Motherboard-Shuttle MK40V	1	$0	$63	$63
	Processor-Duron 1GHz	1	$0	$53	$53
	RAM-SanDisk 512MB cmptflash	1	$0	$39	$39
	DC-DC power converter pw-200-v	1	$0	$50	$50
Batteries	12V Xtreme Plus	2	$0	$38	$38
	18V Milwaukee #48-11-2230	2	$0	$0	$139
Chassis	Steel (frame, axles)	12ft	$0	$35	$35
	Fasteners (bolts, nuts, washers, etc)	1	$0	$40	$40
	#40 Sprockets and Chain	4/8ft	$0	$41	$41
	Bearings	12	$0	$26	$26
	Aluminum (channel, plating)	18ft	$6	$6	$6
	Wheels (+1 caster for rear)	2	$160	$180	$180
Motors	18V Mil. Drill	2	$0	$458	$458
Electronics	Victor 883, 24V Speed Controllers	2	$0	$298	$298
	Logitech QuickCam Pro 4000	1	$106	$106	$106
	GPS-RadioShack DigiTraveler	1	$0	$99	$99
	Compass-PNI Vector 2x	1	$0	$50	$50
	inDart Develop. Kit	1	$0	$0	$308
	ST7 Microcontr.	1	$0	$8	$8
	UARTs	1	$18	$18	$18
Totals			$290	$1608	$2055

Table 4: Cost Break-down

11. Conclusion

The IEEE Robot Team has brought together a diverse team of students to design an autonomous robot named Whitespace to compete in the Intelligent Ground Vehicle Competition. The team designed the robot to exceed all specifications while holding to our team goals. The goals we attempted to achieve included expanding student knowledge and involvement, creating of a robot that is safe, reliable, and durable, and to have fun in the process. We feel that we have in fact met these goals.